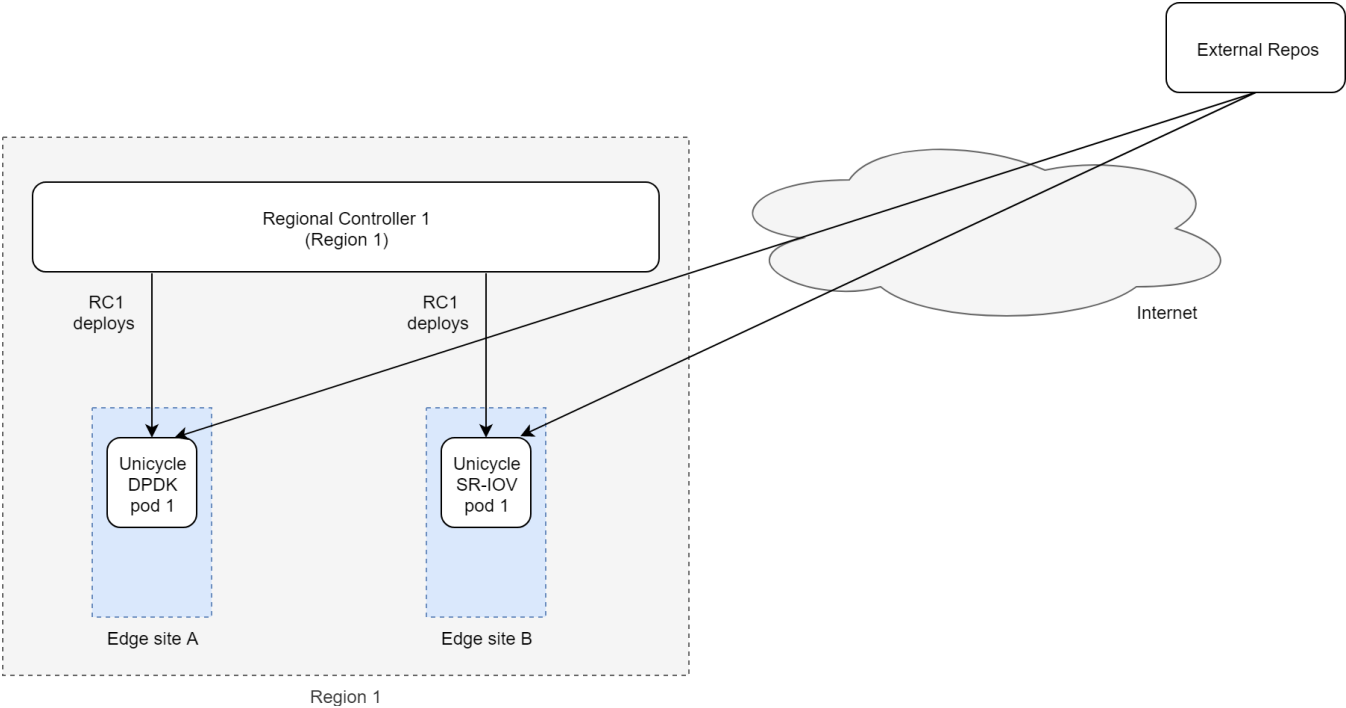


Unicycle Deployment

- [Introduction](#)
 - [Preflight Requirements](#)
 - [Servers](#)
 - [Networking](#)
 - [Software](#)
 - [Generate RC ssh Key Pair](#)
 - [Preflight Checks](#)
 - [Preflight Unicycle Pod Specific Input Data](#)
 - [Unicycle Pods with OVS-DPDK Dataplane](#)
 - [Unicycle Pods with SR-IOV Dataplane](#)
 - [Deploying a Unicycle Pod](#)
 - [Site Specific Input File Selection](#)
 - [Unicycle Pod Site Specific Configuration Input Files](#)
-
- [Example Configuration Input File - Unicycle Pods with OVS-DPDK Dataplane on Dell 740XD Servers](#)
 - [Example Configuration Input File - Unicycle Pods with SR-IOV Dataplane on Dell 740XD Servers](#)
 - [Example Configuration Input File - Unicycle Pods with SR-IOV Dataplane on HP DL380 Gen10 servers](#)

Introduction



Unicycle pods are deployed from an existing Regional Controller and consist of 3 master nodes and 0 to 4 worker nodes. The number of nodes may be increased after validation in subsequent releases.

A number of different options exist including the choice of Dell or HP servers and whether to support tenant VM networking with an OVS-DPDK or SR-IOV data plane. The choice of which is deployed is achieved by simply creating different pod specific yaml input files.

In R1 the options include

Blueprint	Servers	Dataplane	Validated HW details	Validated by
Unicycle	Dell 740XD	OVS-DPDK	Ericsson Unicycle OVS-DPDK Validation HW, Networking and IP plan	Ericsson
Unicycle	Dell 740XD	SR-IOV	ATT Unicycle SR-IOV Validation HW, Networking and IP plan	AT&T
			Ericsson Unicycle SR-IOV Validation HW, Networking and IP plan	Ericsson
Unicycle	HP 380 Gen10	SR-IOV	ATT Unicycle SR-IOV Validation HW, Networking and IP plan	

Preflight Requirements

Servers

Deployment has been validated on two types of servers, Dell and HP.

The exact server specifications used during validation activities are defined in links in the table above. Whilst the blueprint allows for variation in the servers, since the installation includes provisioning of BIOS and HW specific components such as NICs, SSDs, HDDs, CPUs etc only these exact server types are supported and validated in the R1 release.

A unicycle pod deployment consists of at least 3 servers for the master nodes (that can also run tenant VM workloads and 0 to 4 dedicated worker nodes.

All Unicycle pod servers' iDRAC/iLO IP address and subnet must be manually provisioned into the server before installation begins. This includes the genesis, masters and any workers to be deployed in a pod.

Networking

The automated deployment process configures all aspects of the servers including BIOS, linux operating system and Network Cloud specific software. The deployment and configuration of all external switching components is outside the scope of the R1 release and must be completed as a pre-requisite before attempting a unicycle pods deployment.

Details of the networking is given in the [Network Architecture](#) section and that used during the validation in the [ATT Validation Labs](#) and [Ericsson Validation Labs](#) sections of the release documentation.

Software

When a Unicycle pod's nodes are installed on a new bare metal server no software is required on the Target Servers. All software will be installed from the Regional Controller and/or external repos via the internet.

Generate RC ssh Key Pair

In order to ssh into the Unicycle nodes after the deployment process a ssh key pair must be generated on the RC then the public key (located in /root/.ssh/id_rsa.pub) must be inserted into the Unicycle pods's site specific input file.

```
root@regional_server# ssh-keygen -t rsa

.....

root@regional_server# cd /root/.ssh
root@regional_server# ls -lrt
total 12
-rw----- 1 root root 399 May 26 00:34 authorized_keys
-rw-r--r-- 1 root root 395 May 26 00:55 id_rsa.pub
-rw----- 1 root root 1679 May 26 00:55 id_rsa
```

Alternatively, any pre-existing ssh key pair can be used. In that case, the key pair should be manually added to the RC and the corresponding public key should be added to the input file. Instructions on these tasks are beyond the scope of this install guide.

Preflight Checks

To verify the necessary IP connectivity from the RC to the Target Unicycle Server's BMC confirm from the RC that at least port 443 is open to the Target Unicycle Server's iDRAC/iLO BMC IP address. Check the genesis, master and worker node oob ip addresses:

```
root@regional_controller# #nmap -sS <SRV_00B_IP>

root@regional_controller# nmap -sS 10.51.35.145

Starting Nmap 7.01 ( https://nmap.org ) at 2018-07-10 13:55 UTC Nmap scan report for 10.51.35.145 Host is up
(0.00085s latency). Not shown: 996 closed ports PORT STATE SERVICE 22/tcp open ssh 80/tcp open http 443/tcp
open https 5900/tcp open vnc Nmap done: 1 IP address (1 host up) scanned in 1.77 second
```

Note: The enumerated IP shown (10.51.35.146) is an example iDRAC address for a RC deployed to a server.

Report Iss

Preflight Unicycle Pod Specific Input Data

The automated deployment process configures the new Unicycle nodes based on a set of user defined values specific to each Unicycle pod. These values must be defined and stored in a site and pod specific input configuration file before the Unicycle pod deployment process can be started. The input file defines general information such as server names, network, and storage configuration as well as the more complex configuration details like SR-IOV or OVS-DPDK settings.

Unicycle Pods with OVS-DPDK Dataplane

No additional specific preflight checks are required.

Unicycle Pods with SR-IOV Dataplane

No additional specific preflight checks are required.

Deploying a Unicycle Pod

Deployment of each new Unicycle pod at a given site is performed from the RC's UI.

Warning! Internet Explorer and Edge may not work thus it is strongly recommended to use Chrome.

If an action appears to fail click 'Refresh'.

AKRAINO

EDGE STACK

Home

Documentation

Sign Out

Akraino Sites

Akraino Blueprint

ETE Testing

Akraino Sites

Select Regions:

All Sites

Search Edge Sites

Refresh

Build

Deploy

VNF Onboard

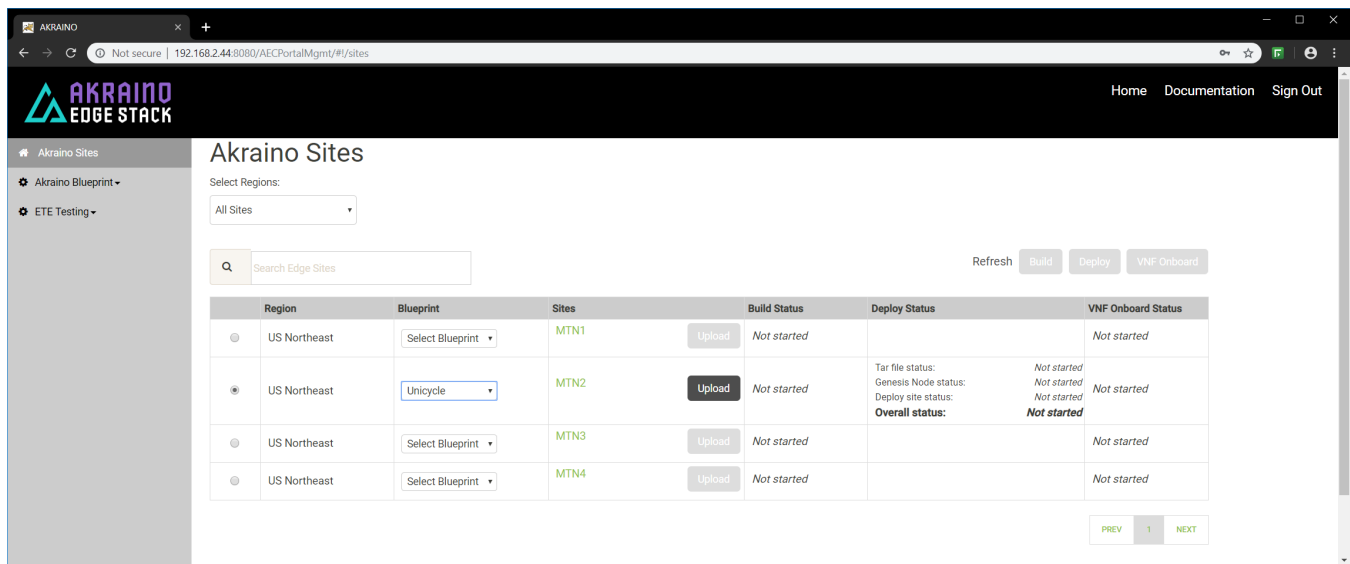
Region	Blueprint	Sites	Build Status	Deploy Status	VNF Onboard Status
US Northeast	Select Blueprint	MTN1	Upload	Not started	Not started
US Northeast	Select Blueprint	MTN2	Upload	Not started	Not started
US Northeast	Select Blueprint	MTN3	Upload	Not started	Not started
US Northeast	Select Blueprint	MTN4	Upload	Not started	Not started

PREV

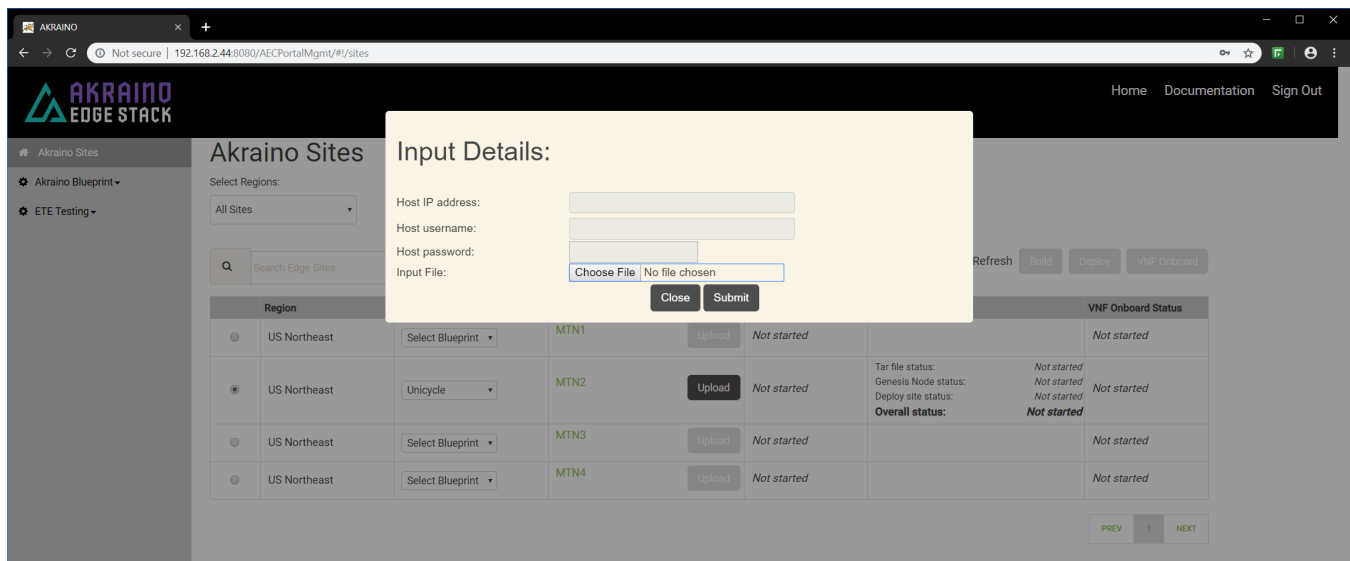
1

NEXT

Select a Region and select 'Unicycle' for the blueprint :



Click 'Upload' to allow you to select the Unicycle site and pod specific input file:



Site Specific Input File Selection

Choose the input file that you have created for the new Unicycle pod you want to deploy then click 'Submit'.

Akraino Sites

Select Regions: All Sites

Search Edge Sites

Refresh Build Deploy VNF Onboard

Region	Blueprint	Sites	Build Status	Deploy Status	VNF Onboard Status
US Northeast	Select Blueprint	MTN1	Upload	Not started	Not started
US Northeast	Unicycle	MTN2 View input File	Upload	Not Started	Not Started
US Northeast	Select Blueprint	MTN3	Upload	Not started	Not started
US Northeast	Select Blueprint	MTN4	Upload	Not started	Not started

PREV 1 NEXT

Click 'BUILD' to create the Airship files necessary to deploy the Unicycle site. Then click 'Refresh' and select the site again:

Akraino Sites

Select Regions: All Sites

Search Edge Sites

Refresh Build Deploy VNF Onboard

Region	Blueprint	Sites	Build Status	Deploy Status	VNF Onboard Status
US Northeast	Select Blueprint	MTN1	Upload	Not started	Not started
US Northeast	Unicycle	MTN2 View input File	Upload	success view yaml build file	Not Started
US Northeast	Select Blueprint	MTN3	Upload	Not started	Not started
US Northeast	Select Blueprint	MTN4	Upload	Not started	Not started

PREV 1 NEXT

Make sure the pod you intend to deploy is still selected.

To initiate the automated deployment click 'Deploy'. You should see the status change to 'In Progress'.

AKRAINO

Home

Documentation

Sign Out

Akraino Sites

Akraino Blueprint

ETE Testing

Akraino Sites

Select Regions:

All Sites

Search Edge Sites

Refresh Build Deploy VNF Onboard

Region	Blueprint	Sites	Build Status	Deploy Status	VNF Onboard Status
US Northeast	Select Blueprint	MTN1	Upload	Not started	Not started
US Northeast	Unicycle	MTN2 View input File	Upload	success view yaml build file	Tar file status: Genesis Node status: Deploy site status: Overall status: In Progress Not Started Not Started In Progress
US Northeast	Select Blueprint	MTN3	Upload	Not started	Not started
US Northeast	Select Blueprint	MTN4	Upload	Not started	Not started

PREV

1

NEXT

The deployment will take a couple of hours to complete. Click "Refresh" periodically to display the current status of each step major step in the deployment.

It is also possible to follow the progress deployment in a number of ways including from the logs on the RC, monitoring the DHCP server on the RC and the virtual console of the iDRAC/iLO on the Genesis server being deployed. Details are shown in the [Deployment Walk-Throughs](#) section of the release documentation.

One of the simplest and easiest approaches is to ssh into the RC's 'host' address and issue the following command:

Report Iss

```
root@regional_controller# tail -f /var/log/akraino/scriptexecutor.log
```

```
2019-05-29 18:35:50.902 DEBUG 18222 --- [SimpleAsyncTaskExecutor-34] .a.b.s.i.DeployResponseSenderServiceImpl :
BuildResponse [siteName=MTN2, buildStatus=null, createTarStatus=null, genesisNodeStatus=null,
deployToolsStatus=null, deployStatus=exception: problem while executing the script. exit code :1,
onapStatus=null, vCDNStatus=null, tempestStatus=null]
2019-05-29 18:35:50.910 DEBUG 18222 --- [SimpleAsyncTaskExecutor-34] .a.b.s.i.DeployResponseSenderServiceImpl :
Build response HttpStatus :200
2019-05-29 18:38:21.965 DEBUG 18222 --- [http-nio-8073-exec-9] c.a.b.controller.CamundaRestController :
Request received for deploy Deploy [sitename=MTN2, blueprint=Rover, filepath=/opt/akraino/redfish
/install_server_os.sh , fileparams=--rc /opt/akraino/server-build/MTN2 --skip-confirm, winscpfilepa, port=22,
username=root, password=akraino,d, destdir=/opt, remotefilename=akraino_airship_deploy.sh,
remotefileparams=null, deploymentverifier=null, deploymentverifierfileparams=null, noofiterations=0,
waittime=0, postverificationscript=null, postverificationScriptparams=null]
2019-05-29 18:38:21.968 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.d.DeployScriptExecutorTaskDelegate :
task execution started :/opt/akraino/redfish/install_server_os.sh --rc /opt/akraino/server-build/MTN2 --
skip-confirm
2019-05-29 18:38:21.968 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
Executing the script.....
2019-05-29 18:38:21.972 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
logging to /var/log/akraino/install_server_os_2019-05-29T18-38-21+0000
2019-05-29 18:38:21.973 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
Beginning /opt/akraino/redfish/install_server_os.sh as user [root] in pwd [/opt/akraino/workflow] with home [
/root]
2019-05-29 18:38:21.978 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
Checking for known required packages
2019-05-29 18:38:22.266 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
python/xenial-updates,now 2.7.12-1~16.04 amd64 [installed]
2019-05-29 18:38:22.540 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
xorriso/xenial,now 1.4.2-4ubuntu1 amd64 [installed]
2019-05-29 18:38:22.816 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
sshd/xenial,now 1.05-1 amd64 [installed]
2019-05-29 18:38:23.094 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
python-requests/xenial-updates,xenial-updates,now 2.9.1-3ubuntu0.1 all [installed]
2019-05-29 18:38:23.332 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
python-pip/xenial-updates,xenial-updates,now 8.1.1-2ubuntu0.4 all [installed]
2019-05-29 18:38:23.579 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
python-yaml/xenial,now 3.11-3build1 amd64 [installed]
2019-05-29 18:38:23.817 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
python-jinja2/xenial,xenial,now 2.8-1 all [installed]
2019-05-29 18:38:24.080 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
make/xenial,now 4.1-6 amd64 [installed,automatic]
2019-05-29 18:38:24.318 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
gcc/xenial,now 4:5.3.1-1ubuntu1 amd64 [installed]
2019-05-29 18:38:24.556 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
coreutils/xenial-updates,now 8.25-2ubuntu3~16.04 amd64 [installed]
2019-05-29 18:38:24.556 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
Tools are ready in [/opt/akraino]
2019-05-29 18:38:24.560 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
Using Build Web ip address [10.51.34.231]
2019-05-29 18:38:34.570 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
WARNING: Preparing to build server [aknode29] using oob ip [10.51.35.146]. Beginning in 10 seconds .....
2019-05-29 18:38:34.572 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
Beginning bare metal install of os at Wed May 29 18:38:34 UTC 2019
2019-05-29 18:38:34.606 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
Found build web ip address [10.51.34.231] on this server!
2019-05-29 18:38:34.608 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
bond0.408 Link encap:Ethernet HWaddr 3c:fd:fe:d2:5b:21
2019-05-29 18:38:34.608 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
: inet addr:10.51.34.231 Bcast:10.51.34.255 Mask:255.255.255.224
2019-05-29 18:38:34.608 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
Checking access to server oob ip [10.51.35.146]
2019-05-29 18:38:36.647 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
64 bytes from 10.51.35.146: icmp_seq=1 ttl=63 time=0.539 ms
2019-05-29 18:38:36.647 DEBUG 18222 --- [SimpleAsyncTaskExecutor-35] c.a.b.s.impl.ScriptExecutionServiceImpl :
64 bytes from 10.51.35.146: icmp_seq=2 ttl=63 time=0.504 ms
```

Unicycle Pod Site Specific Configuration Input Files

This section contains links to the input files used to build the Unicycle pods in ATT's and Ericsson's validation labs for the R1 release. Being pods and site specific the enumerated values will differ. Full details of the relevant validation lab setup that should be referenced when looking at these files is contained in the [Validation Labs](#) section of this documentation.

Please note, superficially these files may appear very similar but they are all included as examination of the details shows the differences due to HW differences such as vendor, slot location of NICs and the method of defining the pod to implement a OCS-DPDK or SR-IOV dataplane as well site specific differences due to VIDs, subnets etc.

- [Example Configuration Input File - Unicycle Pods with OVS-DPDK Dataplane on Dell 740XD Servers](#)
- [Example Configuration Input File - Unicycle Pods with SR-IOV Dataplane on Dell 740XD Servers](#)
- [Example Configuration Input File - Unicycle Pods with SR-IOV Dataplane on HP DL380 Gen10 servers](#)